Hierarchical Up/Down Routing Architecture for Ethernet backbones and campus networks

Guillermo A. Ibáñez [1], Alberto García-Martínez [2], Juan A. Carral [1], Pedro A. González[1],
Arturo Azcorra [2], José M. Arco[1],

1- Universidad de Alcalá, 2- Universidad Carlos III de Madrid

Abstract— We describe a new layer two distributed and scalable routing architecture. It uses an automatic hierarchical node identifier assignment mechanism associated to the rapid spanning tree protocol. Enhanced Up/Down mechanisms are used to prohibit some turns at nodes to break cycles, instead of blocking links like the spannning tree protocol does. The protocol performance is similar or better than other Turn Prohibition algorithms recently proposed with lower complexity *O (Nd)* and better scalability. Simulations show that the fraction of prohibited turns over random networks is less than 0.2. The effect of root bridge election on the performance of the protocol is limited both in the random and regular networks studied.

The use of hierarchical, tree-descriptive addresses simplifies the routing. and avoids the need of all nodes having a global knowleddge of the network topology. Routing frames through the hierarchical tree at very high speed is possible by progressive decoding of frame destination address, without routing tables or port address learning. Coexistence with standard bridges is achieved using combined devices: bridges that forward the frames having global destination MAC addresses as standard bridges and frames with local MAC frames with the proposed protocol.

***Index Terms*—**
Routing, computer networks, protocols, Up/Down routing, turn prohibition, cycle breaking.

I. Introduction

Ethernet is widely adopted at backbones and campus networks due to its excellent price/performance ratio and its configuration convenience. However, the spanning tree protocol limits severely the scalability and performance of Ethernet networks because it blocks all links exceeding the number of network nodes minus one. A new concept of Ethernet switch is needed that combines the features of standard bridge and routers avoiding the limitation of the spanning tree protocol.

Turn Prohibition (*and Up/Down routing)* is a candidate to replace spanning tree in switched networks [3][6]. Spanning tree is used to break cycles in Ethernet networks. Breaking cycles in Ethernet has two reasons. The first is that Ethernet frame does not have a field (like IP has) Time-to-Live (TTL) field to prevent continuous circulation of broadcasted packets in the network. Bridges broadcast packets with unknown destination, multicast and broadcast destination through all bridge ports. The second reason is to prevent deadlocks produced by the IEEE 802.3x flow control mechanism in duplex mode links. If Pause messages are sent through a path with a cycle, all switches stop transmitting. The standard solution for loop prevention is the spanning tree protocol, that builds a tree and blocks the links not belonging to the tree.

Better network infrastructure utilization is possible using VLANs for obtaining multiple spanning trees, or additional encapsulations such as in RBridges [8], Provider Bridges or Provider Backbone Bridge [7], although at the cost of high configuration complexity and/or high resource consumption (protocol specific exchanges, state required, etc).

We investigate a new protocol Hierarchical Up/Down Routing Architecture (HURP) for hierarchical routing over Ethernets using automatically assigned hierarchical, topologically-significant local MAC addresses.

The protocol combines the standard Rapid Spanning Tree protocol for address assignment with a hierarchical distance vector protocol that exchanges routes between neighbor bridges to establish shortest path routes. The layer two frame format lacks a Time To Live (TTL) field. The protocol uses the turn prohibition paradigm to break cycles.

Turn prohibition algorithms define a turn *(a,b,c)* around a node *b* as the pair of links that join *b* with other two nodes like *a* and *c.* When the turn *(a,b,c)* is prohibited, packets arriving at node *b* from link *a-b* can not be forwarded to link *b-c.* All the cycles in a network can be prevented by prohibiting a set of turns among the total turns possible in the topology. Therefore, a limited number of turns guarantee loop free topologies without blocked network links. Networks containing standard bridges may connect to

978-1-4244-2219-7/08/$25.00 (c)2008 IEEE

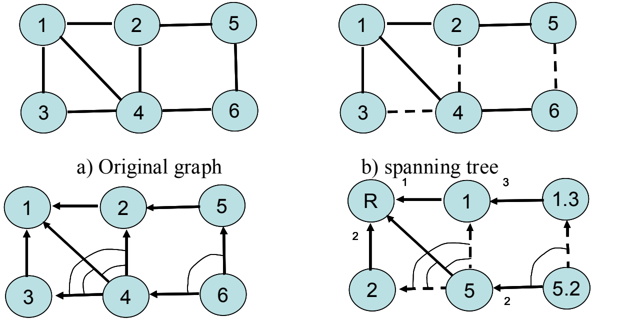
the network via frame encapsulation and other compatibility mechanisms. The main contribution of HURP consists in combining this algorithm with routing based on a hierarchical addressing model built upon the automatic assignment of addresses belonging to the local address space of Ethernet. This assignment is based on the spanning tree. Each bridge is assigned a hierarchical Local MAC address consisting of the chain of port ids of designated ports traversed from root bridge to the root port of that bridge as described below.

II. UP/DOWN AND TURN PROHIBITION

In this section we describe the basis of UP/Down routing and turn prohibition algorithms, which is seminal for the definition of HURP, using the model described in [3].

Consider a network modelled as a directed graph composed of nodes and links. A pair of *(n1, n2)* describes a link from node *n1* to node *n2.* All links are bidirectional. The *degree* of a node is the number of links connecting the node to neighbour nodes. A *path* is a sequence of nodes successively connected by links so that each two subsequent nodes are connected by a link. In opposition to graph theory, a *cycle* in a path occurs when the first *link* and the last *link* of the path are the same, instead of requiring the first and the last node to be the same. A node may be visited repeatedly without creating a cycle. In figure 1, the path 4-3-1-2-4-6 does not contain a cycle. Node 4 is visited twice, but no link is traversed twice.

A *turn* is defined as a pair of input-output links around a node. The three-tuple *(a,b,c)* represents the turn at node *b* from link *a-b* to link *b-c, in Fig. the turn (3,4,2) is the turn* around from node 2 via node 4 to node 3. Unless otherwise stated the turns are symmetrical by default. Therefore, the turn *(a,b,c)* is identical to turn *(c,b,a).* The total number of possible turns around a node of degree *d* is $d•(d-1)/2$. To show the effectiveness of selective prohibition of turns to prevent cycles, prohibiting the turn 2-4-3, prevents cycles in 1-2-3-4



a) Original graph

b) spanning tree

Figure 1 c) Up/Down

d) Up/Down with hierarchical addresses

Turn Prohibition [3] algorithm is centralized and has computational complexity $O(N^2d)$, that limits scalability. Turn Prohibition can provide a loop-free topology by eliminating less than 1/3 of the turns. and it shows an average improvement of performance around 10-20 % [3] compared to its predecessor Up/Down routing paradigm [2].

Some proposals have been made to evolve the turn prohibition model. Tree Based Turn Prohibition (TBTP) [6] provides loop-free topologies with an upper bound for prohibited turns that is half of total for any graph and any spanning tree. The improvement is proportional to the node degree. TBTP relies on STP for convergence. TBTP includes a version that is 802.1D backward compatible. It prohibits less than half of turns of. Complexity is polinomial-time and $\mathcal{O}\leftarrow(N^2 * d^2)$ where N is the number of nodes and d the degree of node with max degree in the graph.

Distributed TBTP

TBTP requires global knowledge topology, what limits scalability when network size increases. A distributed version of TBTP (dTBTP) is proposed [6] to improve scalability, although performance results are slightly inferior to that of TBTP.

III. ADDRESS ASSIGNMENT AND LABELING OF NODES

Labeling nodes with identifiers is used by protocols based on Up/Down routing to assign direction to links. The above mentioned protocols assign identifiers to nodes according to distance to the root bridge.

In [4] a protocol was proposed to assign variable length hierarchical addresses to bridges based on Rapid Spanning Tree Protocol (Fig. 1b). By restricting the maximum address length to 46 bits we can use these addresses as local MAC addresses in the SA and DA fields of Ethernet frames. These addresses are differentiated from global MAC addresses by their local/global bit set to local. This allows coexistence of standard and HLMAC addresses. Hierarchical Local MAC addresses are used to label nodes sequentially that allow using turn prohibition mechanisms to break cycles [2]. They are also used for distance vector routing and for direct forwarding through the spanning tree links without address learning.

A. Global and local MAC addresses

Bit 1 of byte 0 in the standard MAC addresses format indicates if the address is global or local. HLMAC addresses are locally assigned, so bit 1 is set to 1 accordingly.

B. HLMAC addresses format

The Hierarchical Local MAC of a bridge can be expressed in dotted form *a.b.c...* as the chain of designated port IDs *a, b, c, ...* traversed in the descending path from the Root Bridge till the root port of that bridge.

The format of HLMAC is shown in Fig. 2

Octet Nbr	0	1	2	3	4	5
Binary	01*000101*	10001100	*00110011*	11000011	*00111100*	00000000

Fig. 2 Coding of HLMAC address *5*.140.*51*.195.*60*.--

This format is the *implicit* length format, length for first is 6 bits and 8 bits for the other 5 address fields. First byte from left to right with content *00000000* indicates end of address (field not used). Topologically, it express both the address of bridge and the subtree rooted at that bridge. Alternatively, there are also possible *explicit* length formats, for example using 3 bits of byte 0 (bits 2-4) to encode address width level in bits, in other words "mask width per level". The maximum depth of the address is 6 levels for the default implicit format with 8 bits (up to 255 active ports per switch and up to 63 ports for 4 bits).

C. *Automatic assignment of HLMAC addresses*

The assignment of HLMAC addresses to the bridges is based on the spanning tree topology information distributed by the RSTP Protocol from root node to designated nodes. This information is shared with the address assignment mechanism. Additional BPDUs containing HLMAC addresses as shown in figure 2 are interchanged periodically by every bridge with its neighbors assigning the addresses to the bridges connected to their designated ports.

An example is shown in figure 3.The root bridge is the origin of HLMAC addresses, it has no coordinates[1]. The assumption (as in other protocols like RSTP protocol) is that all inter switch links are point to point. Links to hosts (leaf nodes) can be shared. In figure 3, Bridge D1 has 32 as HLMAC address because receives BPDUs from the Root Bridge via Designated Port of ID 32 of Root Bridge. Bridge 32.7 obtains this address because it receives BPDUs from Root Bridge via the Designated Bridge 32 and its Designated Port 7. HLMACs addresses convey the topological position of the bridge in the tree. The default range for port IDs is 0..255. It may be expanded to 10 bits (standard 802.1D port id range) provided equal length is maintained from level 2 of coordinate downwards. The identity (Port ID: 0...1023) of the designated port of each bridge is then used as the topological coordinate of the connected bridge (the bridge attached by its root port to this designated bridge). Note that the second bit from the left is coded to 1 as "local" assigned MAC address. Ports of leaf bridges connected to single host with point to point links may perform translation (NAT) of MAC address to preserve transparency to hosts. Every port needs to store the pairs of global MACs-HLMAC correspondence.

D. *Compatibility with 802.1D*

Ports connected to sub networks of standard bridges encapsulate data frames adding a header with destination address the HLMAC of egress bridge and as source address the ingress bridge HLMAC. At the egress HURP bridge encapsulation is removed and original frame forwarded. NAT of MAC is optional because all user frames can be encapsulated. The overhead is somewhat increased.

Optimizations to maximize the number of levels are possible if any node reports though its root port the maximum degree received from other bridges via designated ports. Root bridge in this case may explicit in the address assignment the mask width selected.

E. *Address length limitations*

When a switch gets assigned an address of maximum depth the HURP protocol dialogues with all the connected switches below in the spanning tree as if they were standard bridges and operates as a HURP edge node working in encapsulation mode. A HURP bridge that can not obtain a valid HLMAC due to address length limitation of their HURP neighbor defaults to standard bridge operation, remaining in HURP passive mode until a valid HLMAC addresses is assigned to him via its root port, indicating that the length restriction has disappeared.

IV. HURP PROTOCOL

HURP (Hierarchical Up/down Routing Protocol). HURP is a hierarchical distance vector routing and forwarding protocol that uses tree-based addresses with Up/Down routing mechanisms to prevent loops. Hierarchical HLMAC addresses allow routing through transversal links, otherwise blocked by the spanning tree protocol, while preventing frame loops just by turn prohibition of the down-up turns. HLMAC addresses may also be used to forward frames to destination along the spanning tree by direct decoding of address prefixes, without neither routing tables nor MAC address learning.

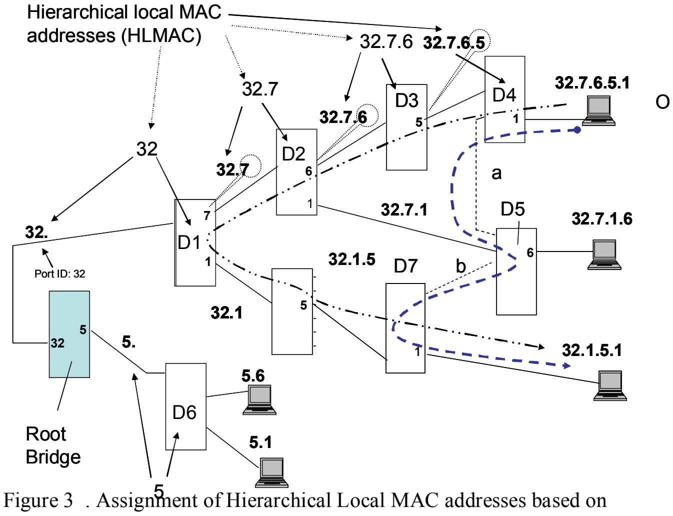


Figure 3 . Assignment of Hierarchical Local MAC addresses based on spanning tree. -..Tree forwarding . --- Transversal forwarding.

HURP Control plane

Apart from the above described HLMAC address assignment and the ancillary RSTP protocol, HURP is a distance vector protocol that interchanges routes between bridges. Every bridge transmits to its neighbours the known shorter distance routes to other bridges that do not create cycles in topology. Legal routes are those that do not use a prohibited turn (down-up) in the node announcing the route.

Operation and messages are similar to RIP protocol with faster interchange intervals (subsecond). The path costs used may be hop counts or standard costs as in 802.1D, inversely proportional to the link speed.

[1] During RSTP topology reconfigurations, the root bridge ID can be considered appended to the HLMAC address, to distinguish an HLMAC from another one based in a different root bridge.

HURP protocol uses transversal routes via links that do not belong to spanning tree when their cost is equal or better than spanning tree cost. The use of hierarchical identifiers with topological information allows an improvement over the U/D routing protocol: the turn that reaches the destination branch for a frame may be permitted even if it is down-up because reachability is guaranteed once the frame reaches any bridge of the destination branch.

User plane. HURP frame forwarding

HURP protocol forwards frames in two modes: with shortest paths via both tree and non-tree (cross) links using routing tables and trough tree links only ascending and descending by destination address decoding. This mode does not require necessarily routing tables, only needs stable HLMAC addresses. A bridge HLMAC address is stable when its root port is enabled. This occurs with RSTP at the same time that the port transits to forwarding state with the port upwards in the tree. The HURP forwarding algorithm is shown at Fig. 4. It operates as follows: if the destination HLMAC is longer than HLMAC of the bridge being traversed, destination is downwards the tree and output port is the first octet exceeding this bridge HLMAC. The opposite happens if destination HLMAC is shorter, frame is forwarded via root port. It does not require MAC address learning as a transparent bridge.

Forwarding in shortest path mode uses the routing tables constructed with the interchange of distance vectors. Routing may be performed on an exact match of destination address or on a prefix matching basis. This allows the aggregation of routes and shortens routing tables.

Fig. 4 shows, with a discontinuous line, the route followed by a frame from an originating terminal O with HLMAC address 32.7.6.5.1 till destination host F with address 32.7.1.5.0. First leaf bridge 32.7.6.5 has a shortest path route through intermediate bridge 32.7.6.1 and forwards the frame through cross links *a* and then *b* till destination F 32.1.5

```
HURP forwarding algorithm
 # Check if destination HLMAC and this bridge HLMAC have some prefix in common (i.e belong to same branch)
    CommonAddressPrefix= CommonPrefix(DestinatAddressHLMAC, ThisBridge.HLMAC)
    IF CommonAddressPrefix ≠ null ; destination host is connected at same tree branch.
    AddressSuffix = DestinationAddress.HLMAC XOR ThisBridge.HLMAC
        IF length (DestinationAddress.HLMAC) < length (ThisBridge.HLMAC); destination host is up in the tree
          output interface = root port   ; root port is selected, to forward frame up in the spanning tree
        FI
        IF length (DestinationAddress.HLMAC) > length (ThisBridge.HLMAC); destination host is down the tree
          forwarding port = (DestinationAddress.HLMAC XOR ThisBridge.HLMAC) AND OctectMask ;
                forward frame via the designated port, coded in 1st octet of suffix
        FI
        IF length (DestinationAddress.HLMAC) = length (ThisBridge.HLMAC); destination is this bridge
          Extract frame ;   tear off HLMAC header and CRC to decapsulate original frame
          Deliver frame ; deliver 802.1D frame to standard bridges.
        FI

    Else ; destination host is in a different branch
        IF ForwardingTable (destination) =/ null ; there is a route (via cross links or tree links)
        output interface = ForwardingTable (DestinationAddress.HLMAC) ; route frame via table
    Else ; there is no HURP route, frame is routed via spanning tree.
      output interface = root port   ; forward frame via root port of this bridge, up in the spanning tree
    FI
```

Fig. 4. HURP forwarding algorithm

HURP Turn Prohibition enhancement

HURP provides an improvement over Up/Down Turn Prohibition made possible by the topology information contained in HLMAC addresses. The basic concept is to permit to frames the turns that end at the destination node or that reach the destination branch of the spanning tree. This is possible because, once a frame arrives at any point of the destination branch, it is guaranteed that the frame will reach the destination just descending or even ascending in the tree. The improvement takes place at the user plane, a frame that is being forwarded at the last hop -1 may be routed. At every bridge, the destination HLMAC is checked to verify if any neighbor is a prefix or contains the destination HLMAC. If so, the frame may be directly forwarded irrespective of turn prohibitions.

A. Compatibility with standard bridges

Several mechanisms for interoperability with standard 802.1D bridges and self configuration have been devised.

The basic compatibility mechanisms used are: encapsulation of frames entering the HURP core from 802.1D bridge subnets, combined devices HURP +802.1D bridges as shown in Fig. 5 with local/global MAC address space separation and automatic construction by all interconnected HURP bridges of a network core using an extended spanning tree protocol protocol that builds a hierarchical spanning tree with all connected combined bridges which act as root bridges of subtrees of standard bridges. The description is out of the scope of this paper.

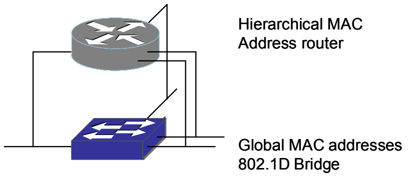


Figure 5. Combined functionality of HURP and standard 802.1D bridge

B. Reconfiguration

A link or bridge failure may cause reconfiguration in the spanning tree and result in changes in active topology. It follows similar rules than RSTP and produces the same effects regarding port states. The main difference is that in RSTP the learnt MAC addresses are flushed while in HURP the assigned HLMACs are deleted from tables. Forwarding based on HLMACs is inmediately stopped at ports that lose their valid HLMAC address until new HLMAC addresses are assigned (i.e. until spanning tree branches are reconfigured). Forwarding of frames with global addresses follows 802.1D rules.

HLMAC addresses might appear as volatile due to their dependency of the spanning tree. However it must be taken into account that the root bridge is carefully chosen so that topology changes are minimized. Besides this, the fast response of RSTP on reconfiguration minimizes unavailability of HLMAC routing. In case of reconfiguration, the topology change notification mechanism of RSTP is used to erase all vector distance routing tables.

V. EVALUATION

A. *Fraction of Prohibited Turns.*

1) *Regular topologies*

We evaluated the fraction of prohibited turns both in regular and random topologies. Regular topologies evaluated are two and three dimensional meshes and hypercube topologies. Square mesh topologies were evaluated of sizes in the range from 16 node (4 x 4) mesh to 144 node (12 x 12) mesh. Hypercube topologies of 8 to 128 nodes were evaluated with uniform node degree increasing from 3 (2^3 = 8 nodes) to 7 (2^7 =128 nodes). The fraction of prohibited turns for a pure U/D based HURP protocol is *constant and independent* of network size: 0.167 for meshes and 0.25 for hipercubes *(2ary n-cubes)*. This higher value is due to the higher node degrees of hypercube topologies. The fraction of prohibited turns for spanning tree in meshes grows from 0,69 for 16 nodes to 0,79 for 144 nodes. There is a moderate dependency on the bridge elected as root (0,57 minimum and 0,69 maximum for the 16 node mesh). HURP U/D Turn Prohibition offers an improvement up to 372% over STP. The fraction of prohibited turns for spanning tree in hypercubes goes from 0,77 for 16 nodes and approaches unity (0,91) for the 128 node hypercube. The fraction of prohibited turns by U/D TP stays fairly constant with increasing network size.

A number of regular tridimensional meshes, ranging from x 3 x 3 to 6 x 6 x 6 nodes was also simulated, selecting every node as root each time. Introducing the additional improvement to HURP of allowing the last turn to destination branch, a significant improvement is obtained over the already low value of Up/Down. Results are shown in Fig. 6 comparing U/D with HURP. U/D results are shown by the upper line (where max, min and average fraction of prohibited turns coincide). HURP average max, average and average min are represented by the three lower lines. The dependency of root bridge election is low, always close to the average value as shown by the HURP max and HURP min lines.

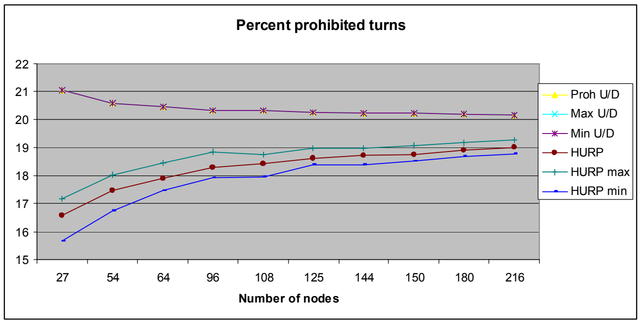


Fig. 6 Percentage of prohibited turns for tridimensional meshes (3x3x3 to 6x6x6 nodes)

2) *Turn prohibition random 120 node topologies*

We evaluated a series of 120 node random topologies with varying average node degree. For each average node degree 40 topologies were generated with BRITE [10] (AS level, Waxman model and default parameter values) and evaluated. Table IV shows the results for Up/Down and HURP. Fraction of prohibited turns increases with node connectivity, but stays under 0,20 always lower than Up/Down and far from STP, that is around 0,95 for degree 8.

TABLE IV
FRACTION OF TURNS PROHIBITED BY EACH ALGORITHM OVER 120 NODE RANDOM TOPOLOGIES AS A FUNCTION OF AVEROGE NODE DEGREE OF TOPOLOGY

average node degree	U/D	HURP U/D
4	0,148	0,139
6	0,185	0,178
8	0,202	0,193

3) *Varying size fixed node degree topologies.*

A number of random topologies with constant node degree of different sizes was also generated and evaluated through simulation. Table V shows the results. Prohibited turns fraction is higher, but HURP keeps performing better than U/D. Performance of U/D in this very specific type of networks tends to the value of regular networks of equivalent degree.

TABLE V
FRACTION OF TURNS PROHIBITED FOR RANDOM TOPOLOGIES FIXED NODE DEGREE (ALL NODES DEGREE FOUR)

nodes	U/D	HURP
16	0,27	0,20
32	0,26	0,21
64	0,25	0,22
128	0,25	0,23

The advantage of HURP over U/D diminishes with increasing node network size due to the lower relative importance of the last turn permission.

B. *Throughput*

In this section we compare the maximum throughput obtained with the algorithms studied. To measure the throughput, we assume that every link client host establishes a session (flow) with C different clients at other nodes. We use a flow model to determine throughput. It is defined as the maximum rate at which each client saturates the most loaded link in the network. We use for all simulations *C=4* so that every bridge supports 4 client sessions with 4 different randomly selected bridges uniformly distributed. With the help of Omnet simulator the *bottleneck link* is determined, that is, the link shared by more sessions. The throughput value is obtained dividing the bottleneck link capacity by the number of sessions that share the link.

HURP provides a great improvement in the throughput compared with STP, due to the use of links otherwise blocked by the spanning tree protocol. The relative improvement is closely related with the node degree because STP limits the active topology to an average degree close to 2 (*2*(N-1)* links divided by *N* nodes).

Fig. 7 shows the comparison of throughput of hypercube topologies with Shortest Paths, Spanning Tree and HURP protocol using *k* (explained below) values 1 and 0,5. HURP with k =1 performs very close (lines coincide in graph) to shortest path regarding throughput.

Improving throughput through load distribution: the k factor

A simple way of distributing traffic away from the spanning tree has been evaluated. It is based on lowering the apparent cost of alternative paths. HURP algorithm uses a configurable *k* factor to reduce the apparent cost of any cross link, increasing the probability of a transversal path being elected instead of the default path via spanning tree. This factor, with range *0 < k <1*, reduces the link costs of transversal paths when compared with the default path cost via spanning tree. A value of *k =1* (default) is neutral in path costs. With *k = 0.5* throughput is improved up to 67% more than with shortest path routing due to better network utilization. This is due to the fact that with values of K lower than unity, longer paths outside the spanning tree appear as shorter than spanning tree paths and traffic gets more dispersed across routes. However this improvement does not extrapolate to all types topologies, it seems limited to networks with average node degrees in a roughly estimated range of 3 to 5.

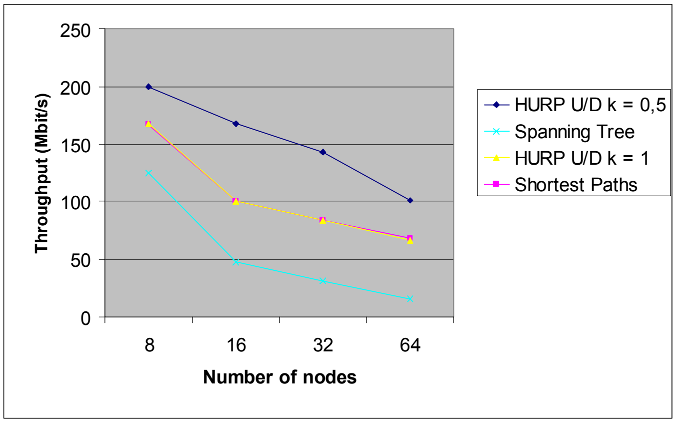


Fig. 7 Comparison of throughput for hypercube topologies

The main advantages provided by HURP are the following: First, it breaks the cycles in the network without risk of network partition. This is derived from the fact that turn prohibition is based on identifiers assigned according to distance to root bridge and topological significance. Second, it operates separately in the local addressing plane of 802.1D so it may coexist in the same device with the standard 802.1D functionality. Third is that computational complexity is that of a distance vector protocol *O(Nd)*, much lower than proposals that require that nodes have global topology knowledge. Fourth is that it can be used with weighted graphs, using for instance standard 802.1D link costs instead of unit cost per hop. The protocol performance is similar or superior to the Up/Down and other Turn Prohibition algorithms, yet it provides higher simplicity and scalability due to the topology information carried on the hierarchical MAC addresses. Performance in all aspects is much closer to shortest path routing than to the spanning tree because the restrictions to routing are much less.

VI. Conclusion

We have described a self configuring hierarchical frame routing architecture that uses turn prohibition to prevent cycles and deadlocks. It can be implemented as either stand alone or combined in the same device with standard 802.1D bridges coexisting in a network of hierarchical routing bridges. Mechanisms for coexistence will be described in detail in other paper. Evolution is possible via software migration. Routing mechanism is simple. Forwarding through the spanning tree is suitable for very high speeds switches due to the absence of routing tables or port cachés. HURP routing, as other turn prohibition algorithms, is a universal mechanism for preventing packet loops and breaking cycles. It is applicable to LANs like Gigabit Ethernets to prevent packet loops and deadlocks in full duplex and to networks using wormhole routing like Network of Workstations.

References

[1] C. Glass and L. Ni, “The Turn Model for Adaptive Routing,” Journal of ACM, Vol. 41, No. 5, pp. 874–902, September 1994.

[2] Shoreder, M. et al.: Autonet: A High-Speed, Self–Configuring Local Area Network Using Point –to–Point Links. IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, pp. 1318–1335, October 1991.

[3] Starobinski, D.; Karpovsky, G.; Zakrevsky, F.: Applications of Network Calculus to General Topologies, IEEE/ACM Transactions on Networking, June 2003, vol 11, No. 3, pp 411-422.

[4] G. Ibáñez, A. Azcorra. Application of Rapid Spanning Tree Protocol for Automatic Hierarchical Address Assignment to Bridges. 11th International Telecommunication Networks Strategy and Planning Symposium. Networks 2004. Wien. June 2004. Available online: www.ieee.org/ieee.explore.

[5] Omnet++: www.omnetpp.org.

[6] Pellegrini et al. Scalable, Distributed cycle-breaking algorithms for gigabit Ethernet backbones. Journal of Optical Networking, Vol. 5. No. 2., Feb. 2006

[7] IEEE 802.1 Working group. http://www.ieee802.org/1

[8] R. Perlman, “Rbridges: Transparent routing,” en Proceedings of IEEE Infocom 2004, March 2004.

[9] IEEE 802.1D-2004 IEEE standard for local and metropolitan area networks-Media access control (MAC) Bridges. http://standards.ieee.org/getieee802/802.1.html.

[10] Boston University Representative Topology Generator - BRITE, availableonline at http://www.cs.bu.edu/brite/.

978-1-4244-2219-7/08/$25.00 (c)2008 IEEE